graphics memory access bus into individual bus partitions, each of which is a fraction of the graphics memory access bus size, said memory controller partitioning information within said graphics memory into independently accessible memory partitions, said memory controller routing data from said independently accessible memory partitions to said plurality of graphics processing units via said individual bus partitions.

33. (New) The graphics system of claim 32 wherein said memory controller includes control logic to select one or more of said individual bus partitions to route data in response to a data request from a graphics processing unit of said plurality of graphics processing units.

34. (New) The graphics system of claim 32 wherein said memory controller maps data to said independently accessible memory partitions in an interleaved fashion to balance memory load across said independently accessible memory partitions.

35. (New) The graphics system of claims 32 wherein said individual bus partitions have corresponding individual queues.

36. (New) The graphics system of claim 35 further comprising a multiplexer to combine data from said individual queues.

37. (New) The graphics system of claim 35 wherein said individual queues have corresponding arbiter circuits.

38. (New) The graphics system of claim 35 wherein said individual queues facilitate ordered read data delivery and thereby prevent deadlocking across said individual bus partitions.

39. (New) The graphics system of claim 38 wherein said individual queues process read data requests that span a plurality of individual bus partitions.

40. (New) The graphics system of claim 37 wherein said arbiter circuits include an arbiter circuit to prioritize requests from a sub-set of said plurality of graphics processing units.

41. (New) The graphics system of claim 40 wherein said sub-set of said plurality of graphics processing units share a command and write data path.

42. (New) The graphics system of claim 40 wherein each graphics processing unit of said sub-set of said plurality of graphics processing units has a sub-request ID.

43. (New) The graphics system of claim 40 wherein said arbiter circuit pre-arbitrates requests from a sub-set of low-bandwidth graphics processing units.

44. (New) The graphics system of claim 43 wherein said arbiter circuit treats said sub-set of low-bandwidth graphics processing units as a single client.

45. (New) The graphics system of claim 32 wherein individual memory partitions of said independently accessible memory partitions are assigned to solely service individual graphics processing units of said plurality of graphics processing units.

46. (New) The graphics system of claim 32 wherein a selected graphics processing unit of said plurality of graphics processing units accepts data in an out-of-order fashion.

47. (New) The graphics system of claim 46 wherein said selected graphics processing unit accepts data as soon as said data is available from a partition.

48. (New) The graphics system of claim 35 wherein said individual queues include request queues and read data return queues to balance data locality requirements to facilitate memory access efficiency.

49. (New) The graphics system of claim 37 wherein each arbiter circuit implements an independent priority policy to route information to said plurality of graphics processing units.

50. (New) The graphics system of claim 49 wherein said priority policy is a static policy.

51. (New) The graphics system of claim 49 wherein said priority policy is a least recently used policy.

52. (New) The graphics system of claim 49 wherein said priority policy is a round-robin policy.

53. (New) The graphics system of claim 49 wherein said priority policy is a fixed priority policy.

54. (New) The graphics system of claim 49 wherein said priority policy is a dynamic priority policy.

55. (New) A method of servicing data requests from graphics processing units, comprising:

receiving data requests from graphics processing units accessing a unitary graphics memory;

assigning said data requests to one or more independently accessible memory partitions imposed upon said unitary graphics memory; and

delivering data from said independently accessible memory partitions to said graphics processing units via individual bus partitions of a unitary graphics memory access bus.

56. (New) The method of claim 55 further comprising storing data from said independently accessible memory partitions prior to said delivering.

57. (New) The method of claim 56 further comprising combining stored data prior to said delivering.

58. (New) The method of claim 57 further comprising facilitating ordered read data delivery to prevent deadlocking across said individual bus partitions.

59. (New) The method of claim 58 further comprising prioritizing requests from a sub-set of said plurality of graphics processing units.

60. (New) The method of claim 59 further comprising sharing a command and write data path between a sub-set of said plurality of graphics processing units.

61. (New) The method of claim 60 further comprising assigning a sub-set request ID to individual graphics processing units of said sub-set of said plurality of graphics processing units.

62. (New) The method of claim 55 further comprising pre-arbitrating requests from a sub-set of low-bandwidth graphics processing units of said graphics processing units.

63. (New) The method of claim 62 further comprising treating said sub-set of low-bandwidth graphics processing units as a single client.

64. (New) The method of claim 55 further comprising assigning individual memory partitions of said independently accessible memory partitions to service individual graphics processing units of said graphics processing units.

65. (New) The method of claim 55 further comprising accepting data at a selected graphics processing unit of said graphics processing units in an out-of-order fashion.

66. (New) The method of claim 65 further comprising accepting data as soon as said data is available from a memory partition.

67. (New) The method of claim 55 further comprising balancing data locality requirements to facilitate memory access efficiency.

REMARKS

Claims 1-31 are rejected in view of ten patents cited by the Patent Examiner. Applicant has canceled claims 1-31 and submits new claims 32-67 to fully distinguish over the prior art of record.